



PLATFORMODE ÖDEME POSTMAN ENTEGRASYON TEKNİK DOKÜMANI

PLATFORM ÖDEME HİZMETLERİ VE ELEKTRONİK PARA A.Ş

İSTANBUL



İÇİNDEKİLER

İÇİNDEKİLER.....	1
1. Giriş	2
1.1. Amaç	2
1.2. Kapsam.....	2
1.3. Erişim URL'leri	2
2. Postman Kurulumu.....	2
2.2. Postman Environment Dosya Yükleme	3
3. Get Token	4
4. Purchase Link	5
5. PaySmart2D.....	7
6. PaySmart3D.....	9
7. Refund API.....	11
8. Checkstatus	12
9. Confirm Payment	13
10. Get Installment.....	14
11. Get Pos	15
12. Merchant Commissions	16
13. Get Transactions	17
14. All Transactions	18
15. Payment Complete.....	19



1. Giriş

Bu belge, Platformode Ödeme Entegrasyonuna yardımcı olması için Postman ile test ortamında API isteklerinin test edilmesi için hazırlanmıştır.

1.1. Amaç

Bu belgenin amacı, çeşitli ödeme seçenekleriyle ödeme işlemlerini gerçekleştirmek için Platformode Ödeme sistemini e-ticaret sistemlerine entegre etmek üzere üye işyerlerine rehberlik etmektir.

1.2. Kapsam

Satıcı uygulamaları ile Platformode ödeme sistemi arasındaki entegrasyonun kapsamı, ödeme entegrasyonu gereksinimi için işlevsel gereksinimi kapsar.

1.3. Erişim URL'leri

Ortama bağlı olarak aşağıdaki erişim URL'leri kullanılacaktır.

Ortam	URL Format
Test Ortamı	https://testapp.platformode.com.tr/ccpayment
Canlı Ortam	https://app.platformode.com.tr/ccpayment

2. Postman Kurulumu

Postman ile test yapılabilmesi için [Platformode-Payment-Gateway-Integration-Samples](#) github linkine Postman klasörüne önceden hazırlanmış aşağıdaki isimleri olan 2 adet Postman json dosyası yüklenmiştir. Ayrıca 3D'li işlemler için de örnek html dosyası oluşturulmuştur. Öncelikle bu üç dosyanın bilgisayara indirilmesi gerekir.

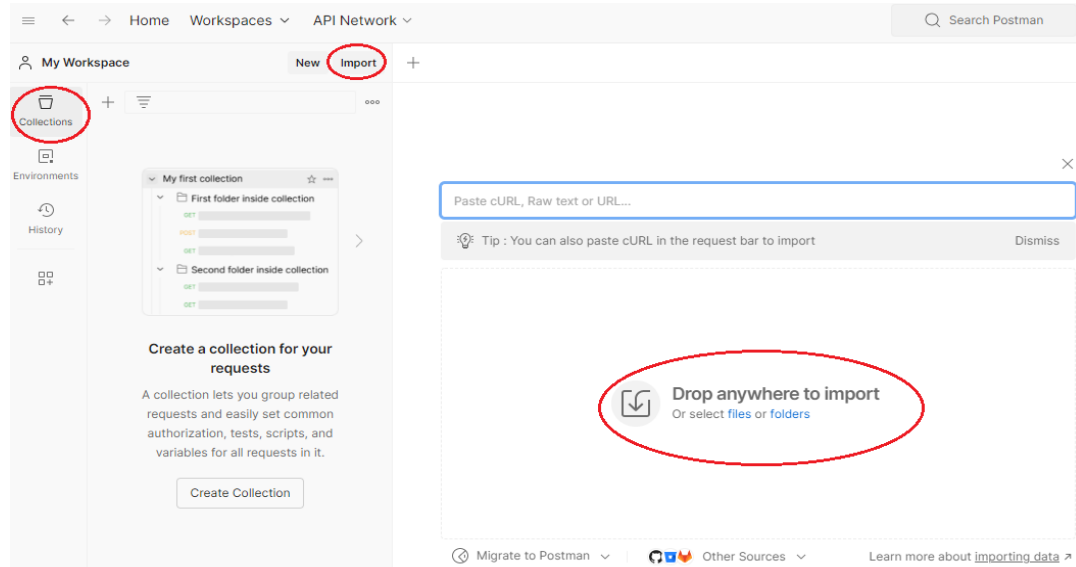
API Environment Dosyası	Platform_Test.postman_environment.json
API Collection Dosyası	Platformode Api Entegrasyonu_Collection_V2_1.postman_collection.json
3D Test Html	3DTestWithCvv.html

2.1. Postman Collections Dosya Yükleme

1- Sol menüde bulunan Collections linkine tıklanır.

2- Import linkine tıklanır.

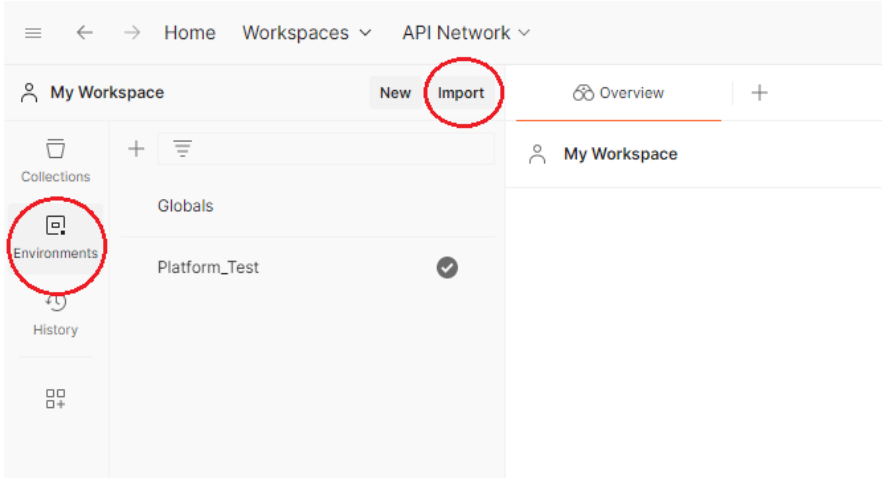
3- Platformode Api Entegrasyonu_Collection_V2_1.postman_collection.json dosyası sürükleyip bırak ile seçilir.





2.2. Postman Environment Dosya Yükleme

- 1- Postman uygulaması açılır.
- 2- Sol menüde bulunan Environments linkine tıklanır.
- 3- Import linkine tıklanır
- 4- Platform_Test.postman_environment.json dosyası sürükleyip bırakarak seçilir.

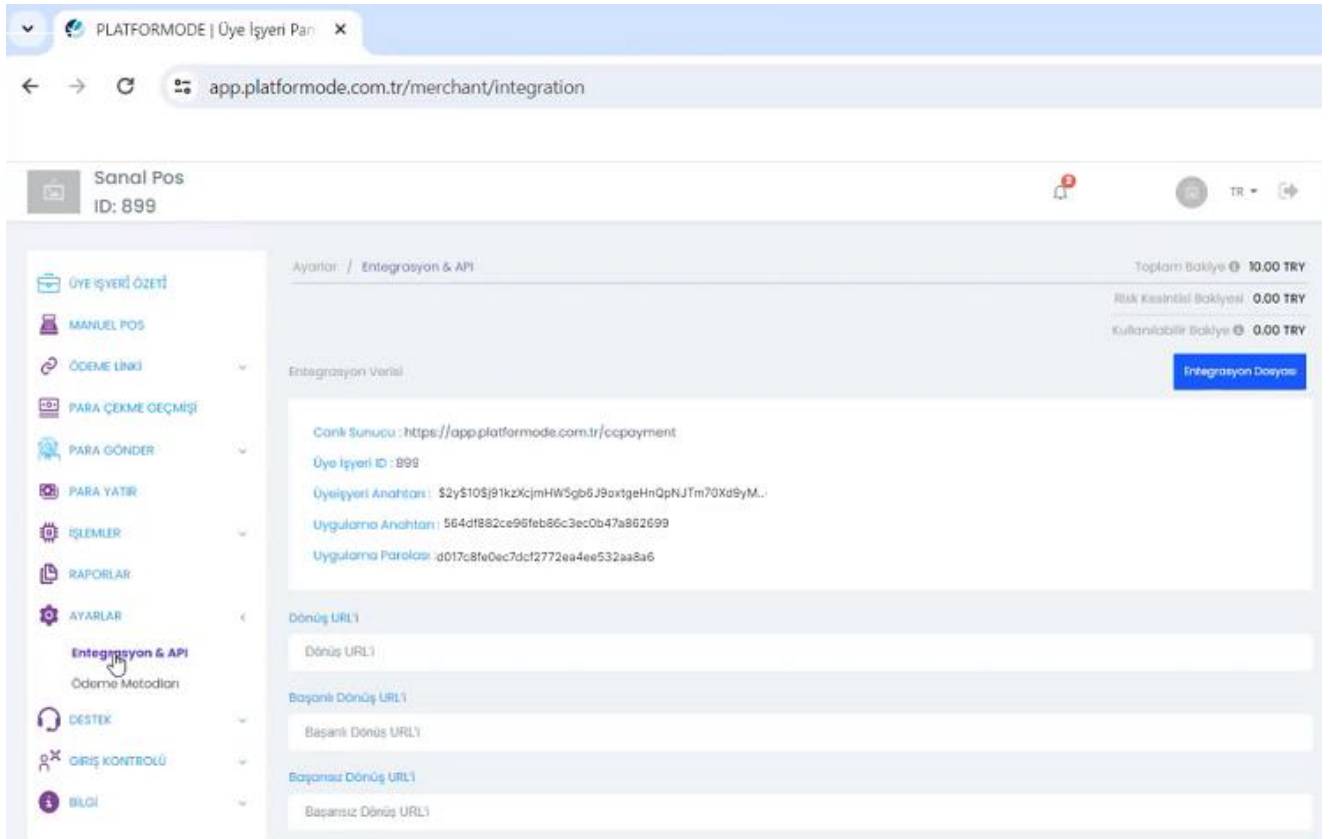


- 5- Üye işyeri panelinden Ayarlar/Entegrasyon&API menüsünden Entegrasyon verileri alınır.

Üye işyeri Anahtarı bilgisi kopyalanır ve Postman Environment bölümünde **merchant_key** alanına,

Uygulama Anahtarı bilgisi kopyalanır ve Postman Environment bölümünde **app_id** alanına,

Uygulama Parola bilgisi kopyalanır ve Postman Environment bölümünde **app_secret** alanına kopyalanır.





4. Purchase Link

Link ile satış için Purchase Link tıklanır. Environment içinde önceden tanımladığınız ve firmanıza ait olan merchant_key ile currency_code, invoice, name, surname bilgileri post olarak gönderilmesi gerekir. Böylece Satış bilgilerini içeren ve link ile ödeme yapacak müşteriye gönderilecek link bilgisi dönecektir.

The screenshot shows the Postman interface for a POST request to the `PlatformOde Api Entegrasyonu / Purchase Link` endpoint. The request body is a JSON object with the following fields:

```
{  "merchant_key": "[merchant_key]",  "currency_code": "TRY",  "invoice": {    "invoice_id": "Abc1234",    "invoice_description": "Testdescription",    "total": 5.00,    "return_url": "https://google.com.tr",    "cancel_url": "https://github.com.tr",    "items": [      {        "name": "Item1",        "price": 5,        "quantity": 1,        "description": "Test"      }    ]  },  "name": "John",  "surname": "Dao"}
```

The response is a JSON object with the following fields:

```
{  "status": true,  "status_code": 100,  "success_message": "Success !",  "link": "https://testapp.plafomode.com.tr/ccpayment/makepayment/VP17133624444682981",  "order_id": "VP17133624444682981"}
```



<https://testapp.platformode.com.tr/ccpayment/makepayment/VP17133624444682901> adresine girildiği zaman aşağıdaki sayfa açılır.

Cep telefonu numarası girildikten sonra, Sistem tarafından girilen cep telefonuna doğrulama kodu gönderilir. Gelen doğrulama kodu girildikten sonra ödemenin alınabilmesi için kredi kartı bilgisi girilerek ödeme tamamlanır.



5. PaySmart2D

2D ile satış işlemi için [PaySmart2D](#) linkine tıklanır.

The screenshot shows the Postman API client interface. On the left, the 'My Workspace' tab is active, and the 'PlatformOde Api Entegrasyonu' collection is expanded. The 'POST PaySmart2D' endpoint is selected. The main area shows the request details for the POST method. The URL is `{{base_url}}/api/paySmart2D`. The request body is a JSON object with the following structure:

```
1 {
2   "cc_holder_name": "Test kart",
3   "cc_no": "4132260000000003",
4   "expiry_month": "12",
5   "expiry_year": "2024",
6   "cvv": "555",
7   "currency_code": "TRY",
8   "installments_number": 1,
9   "invoice_id": "Abc12345",
10  "invoice_description": "INVOICE-TEST-DESCRIPTION",
11  "total": 22.00,
12  "items": [{"name": "Item3", "price": 22, "quantity": 1, "description": "item3 description"}],
13  "name": "John",
14  "surname": "Dao",
15  "merchant_key": "{{merchant_key}}",
16  "hash_key": "{{hash_key}}"
17 }
```

The response body is a JSON object with the following structure:

```
1 {
2   "status_code": 100,
3   "status_description": "Payment process successful",
4   "data": {
5     "order_no": "VP17134385953683872",
6     "order_id": "VP17134385953683872",
7     "invoice_id": "Abc12345",
8     "credit_card_no": "413226****0003",
9     "transaction_type": "Auth",
10    "payment_status": 1,
11    "payment_method": 1,
12    "error_code": 100,
13    "error": "Transaction Successful",
14    "auth_code": 427698,
15    "merchant_commission": 0,
16    "user_commission": 0,
17    "merchant_commission_percentage": 0,
18    "merchant_commission_fixed": 0,
19    "hash_key": "3b48787b115a8aed:7902:WQ4nZdwqB1MCP64euCQHfQCxcFbMegAdoQA8HXgw_I1dwQHAi0t9Gsk+8ug+9qaT",
20  }
21 }
```




Ön Provizyon ile satış yapmak için isteğe "transaction_type":"PreAuth", satırının eklenmesi gerekir.

PlatformOde Api Entegrasyonu

POST Get Token

POST Purchase Link

POST PaySmart2D

POST PaySmart3D

POST Refund API

POST Checkstatus

POST Confirm Payment

POST Get Installment

POST Merchant Commissions

POST Get Pos

POST Get Transaction

POST All Transaction

POST Payment Complete

POST

{{base_url}}/api/paySmart2D

Params Authorization Headers (13) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "cc_holder_name": "Test kart",
3   "cc_no": "4132260000000003",
4   "expiry_month": "12",
5   "expiry_year": "2024",
6   "cvv": "555",
7   "currency_code": "TRY",
8   "installments_number": 1,
9   "invoice_id": "Abcd123456",
10  "invoice_description": "INVOICE TEST DESCRIPTION",
11  "total": 26.00,
12  "items": [{"name": "Item6", "price": 26, "quantity": 1, "description": "item6 description"}],
13  "name": "John",
14  "surname": "Dao",
15  "transaction_type": "PreAuth",
16  "merchant_key": "{{merchant_key}}",
17  "hash_key": "{{hash_key}}",
18 }
```

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status_code": 100,
3   "status_description": "Payment process successful",
4   "data": {
5     "order_no": "VP17135180952054115",
6     "order_id": "VP17135180952054115",
7     "invoice_id": "Abcd123456",
8     "credit_card_no": "413226****0003",
9     "transaction_type": "Pre-Authorization",
10    "payment_status": 1,
11    "payment_method": 1,
12    "error_code": 100,
13    "error": "Transaction Successful",
14    "auth_code": 421575,
15    "merchant_commission": 0,
16    "user_commission": 0,
17    "merchant_commission_percentage": 0,
```

Bu işlemin sonucu Üye İşyeri Panelinden aşağıdaki gibi sonuç izlenebilir.

Detaylar	İşlem ID	Sipariş Numarası	Durum	Kullanıcı Adı	Tutar	Ürün Tutarı	IP	Tarih & Saat	Link ID	GSM Numarası
Detaylar	Sqzv-cg-TC10-37878-190424	VP17135180952054115	Ön Provizyon	-	26.00 ₺	26.00 ₺	193.108.213.144	19.04.2024 - 12:14	0	-
Detaylar	6LdKX-bg-TC10-37878-190424	VP17135090075462952	İade ediliyor	-	25.00 ₺	25.00 ₺	193.108.213.144	19.04.2024 - 09:43	0	-
İade Detayları										

Sayfa başına 10



6. PaySmart3D

3D ile satış işlemi için PaySmart3D linkine tıklanır. Buradan istek gönderilirse başarılı ise başarılı belirlenen Url gidecektir. Dummy banka ekranı görünmeyecek. Dummy banka ekranı için istek buradan göndermek yerine aşağıdaki yol izlenebilir.

The screenshot shows the Postman API client interface. The 'My Workspace' section on the left lists various API endpoints under 'PlatformOde Api Entegrasyonu'. The 'POST PaySmart3D' endpoint is highlighted with a red circle. The main panel shows the 'POST' method with the URL '({base_url})/api/paySmart3D'. The 'Body' tab is selected, displaying a JSON payload for a 3D transaction. The 'Tests' tab is also visible in the sidebar.

```
1 {
2   "cc_holder_name": "Test kart",
3   "cc_no": "4132260000000003",
4   "expiry_month": "11",
5   "expiry_year": "2027",
6   "currency_code": "TRY",
7   "installments_number": 1,
8   "invoice_id": "Test146",
9   "invoice_description": "INVOICE-TEST-DESCRIPTION",
10  "total": 40.00,
11  "items": [{"name": "Item3", "price": 40, "quantity": 1, "description": "item3-description"}],
12  "name": "John",
13  "surname": "Dao",
14  "return_url": "https://www.google.com/",
15  "cancel_url": "https://www.github.com/",
16  "payment_completed_by": "merchant",
17  "transaction_type": "PreAuth",
18  "merchant_key": "{{merchant_key}}",
19  "hash_key": "{{hash_key}}"
20 }
```

PaySmart3D tıklandıktan sonra Tests sekmesine tıklanır. Burada bulunan kodlar seçilir ve kopyalanır.

The screenshot shows the Postman API client interface with the 'Tests' tab selected for the 'POST PaySmart3D' endpoint. The 'Tests' tab is highlighted with a red circle. The main panel displays a C# script for generating a 3D hash key. The script includes comments in Turkish and code for generating the hash key using the 'threeDHashKey' function.

```
1 /*
2
3 Manuel olarak 3d hash key oluşturmak için aşağıdaki web sitesine gidiniz ve tüm kodları yapııştırıp Run butonuna basınız.
4 https://dotnetfiddle.net/
5
6
7 using System;
8 using System.Security.Cryptography;
9 using System.Text;
10 using System.Linq;
11
12 public class Program
13 {
14     public static void Main()
15     {
16         // Lütfen total, installment, currency_code, merchant_key ve invoice_id bilgilerini kontrol ediniz. Aşağıdaki alanları oluşturmak istediğiniz 3d key
17         // bilgilerine göre düzenleyiniz.
18
19         string appSecret = "d817c8f8ec7dcf2772ea4ee532aa8a6"; // app secret
20         string merchant_key = "52y$185j91k2XcjHw5Gb39oxTgeHnQp8J7h78Xd9yMc0jH4Kwjg87ztic"; // merchant_key
21         string total = "10"; // total
22         string installment = "1"; // installment
23         string currency_code = "TRY"; // currency_code
24         string invoice_id = "assd123gba1a"; // invoice_id
25
26         var threeDHashKey = GenerateHashKey(
27             appSecret,
28             total,
29             installment,
30             currency_code,
31             merchant_key,
32             invoice_id
33         );
34
35         Console.WriteLine("3D Hash key için girilen parametreler");
36         Console.WriteLine("appSecret : " + appSecret);
37     }
38 }
```



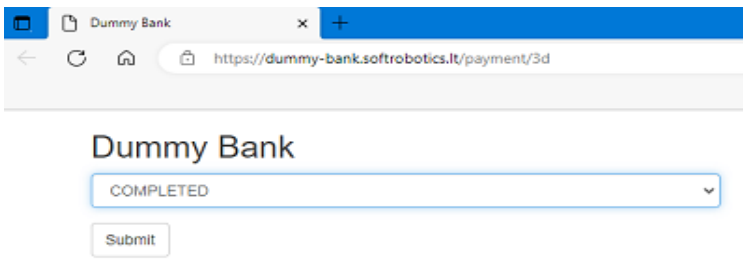
3d Hash Key Bilgisini alabilmek için kopyanan C# kodları <https://dotnetfiddle.net/> sitesine yapıştırılır. appSecret, Merchant_key, total,invoice_id alanları güncellenir ve Run butonuna basılarak çalıştırılır.

```
1 using System;
2 using System.Security.Cryptography;
3 using System.Text;
4 using System.Linq;
5
6 public class Program
7 {
8     public static void Main()
9     {
10         // Lütfen total, installment, currency_code, merchant_key ve invoice_id bilgilerini kontrol ediniz. Aşağıdaki alanları oluşturmak istediğiniz 3d key bilgilerine göre düzenleyiniz.
11
12         string appSecret = "a9b5c28211388c7a6a785122a24e6f46"; // app secret
13         string merchant_key = "$2y$10$e8Mzadv9523RvRqL7K3.efxdERBrbaVZNa3AwlmZriYnrygSH72"; // merchant_key
14         string total = "40"; // total
15         string installment = "1"; // installment
16         string currency_code = "TRY"; // currency_code
17         string invoice_id = "Test146"; // invoice_id
18
19         var threeDHashKey = GenerateHashKey(
20             appSecret,
21             total,
22             installment,
23             currency_code,
24             merchant_key,
25             invoice_id
26         );
27
28         Console.WriteLine("3D Hash key için girilen parametreler");
29         Console.WriteLine("appSecret : " + appSecret);
30         Console.WriteLine("merchant_key : " + merchant_key);
31         Console.WriteLine("total : " + total);
32         Console.WriteLine("installment : " + installment);
33         Console.WriteLine("currency_code : " + currency_code);
34         Console.WriteLine("invoice_id : " + invoice_id);
35
36         3D Hash key için girilen parametreler
37         appSecret : a9b5c28211388c7a6a785122a24e6f46
38         merchant_key : $2y$10$e8Mzadv9523RvRqL7K3.efxdERBrbaVZNa3AwlmZriYnrygSH72
39         total : 40
40         installment : 1
41         currency_code : TRY
42         invoice_id : Test146
43
44         Uzun 3D Hash Key Bilgisi :
45         617364f3bf302ad6:44ca:4tmPfiHoqSg8TBwZ6APyY8uG1zLZSq+474drHwCLPA16rHCADmrZTwtJ2g3HjLUAY___t1SLaV53ekQ1ZgfrnsikCUQUSITIEJobgucSDV7Es
```

3DTestWithCvv.html dosyasında invoice_id, total, items tutar, merchant_key ve 3d Hash Key alanları güncellendikten sonra bu html sayfası browserdan açılır. Dummy Bank doğrulama sayfası gelir.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>3D Test</title>
5 </head>
6 <body>
7 <form name="form" action="https://testapp.platformmode.com.tr/ccpayment/api/paySmart3D" method="POST">
8 <input type="hidden" name="cc_holder_name" value="Test kart">
9 <input type="hidden" name="cc_no" value="4132260000000003">
10 <input type="hidden" name="expiry_month" value="11">
11 <input type="hidden" name="expiry_year" value="2027">
12 <input type="hidden" name="cvv" value="193">
13 <input type="hidden" name="currency_code" value="TRY">
14 <input type="hidden" name="installments_number" value="1">
15 <input type="hidden" name="invoice_id" value="Test146">
16 <input type="hidden" name="invoice_description" value="INVOICE TEST DESCRIPTION">
17 <input type="hidden" name="total" value="40.00">
18 <input type="hidden" name="items" value='[{"name":"Item3","price":40,"quantity":1,"description":"item3 description"}]'\>
19 <input type="hidden" name="name" value="John">
20 <input type="hidden" name="surname" value="Dao">
21 <input type="hidden" name="return_url" value="https://www.google.com/">
22 <input type="hidden" name="cancel_url" value="https://www.github.com/">
23 <input type="hidden" name="payment_completed_by" value="merchant">
24 <input type="hidden" name="merchant_key" value="$2y$10$e8Mzadv9523RvRqL7K3.efxdERBrbaVZNa3AwlmZriYnrygSH72">
25 <input type="hidden" name="hash_key" value="617364f3bf302ad6:44ca:4tmPfiHoqSg8TBwZ6APyY8uG1zLZSq+474drHwCLPA16rHCADmrZTwtJ2g3HjLUAY___t1SLaV53ekQ1ZgfrnsikCUQUSITIEJobgucSDV7Es">
26 </form>
27 <script>
28 type="text/javascript">
29 form = document.forms[0]
30 form.submit();
31 </script>
32 </body>
33 </html>
```

İşlem başarılı olursa tanımlan Başarılı Dönüş URL'ye, başarısız olursa Başarılı Dönüş URL adresine gider.



7. Refund API

İade işlemi yapılacak işleme ait fatura numarası bilinmiyorsa Üye İşyeri Panelinden İşlemler menüsünden ilgili işlem üzerinde Detaylar butonu içinden fatura numarası alınabilir.

İşlemler

Kullanıcı GSM

Minimum Tutar

Maximum Tutar

Ara ...

İşlem ID

Ödeme aracı

Durum

Link ID

Original

Ödeme Türü

Filtreyi Temizle

Kaydet

CSV ~

Dişarı Aktar ▶

Dişarı Aktarılan Rapor Geçmişi

Detaylar	İşlem ID	Sipariş Numarası	Durum	Kullanıcı Adı	Tutar	Ürün Tutarı	IP	Tarih & Saat	Link ID	GSM Numarası
Detaylar	6LdKX-bg-TC10-37878-190424	VPI7135090075462952	Tamamlandı	-	25.00 ₺	25.00 ₺	193.108.213.144	19.04.2024 - 09:43	0	-

Sayfa başına

10

İade işlemi için [Refund API](#) linkine tıklanır. Invoice_id alanına fatura numarası, amount alanına iade edilmek istenen olan tutar yazılır. Fatura tutarının tamamı iade edilebileceği gibi kısmı bir tutar da iade edilebilir. Bir fatura için parça parça da iade işlemi yapılabilir.

The screenshot shows the Postman interface with the 'POST Refund API' endpoint selected. The main panel displays the API details for 'PlatformOde Api Entegrasyonu / Refund API'. The method is 'POST', and the URL is 'POST PaySmart2D'. The body is a JSON object with the following fields:

```

1 {
2   "invoice_id": "Abcd12345",
3   "amount": 25,
4   "app_id": "{{app_id}}",
5   "app_secret": "{{app_secret}}",
6   "merchant_key": "{{merchant_key}}",
7   "hash_key": "{{hash_key}}",
8 }

```

The 'Body' tab is active, showing the JSON content. The 'JSON' tab is also visible, showing the response body:

```

1 {
2   "status_code": 100,
3   "status_description": "Refund completed successfully",
4   "order_no": "VP17135090075462952",
5   "invoice_id": "Abcd12345",
6   "ref_no": "",
7   "ref_number": ""
8 }

```

İade işlemi sonucu Üye İşyeri Panelinden aşağıdaki gibi sonuç izlenebilir.

[illegible]



8. Checkstatus

Ödeme işleminin durumu görmek için Checkstatus linkine tıklanır. Ödeme durumu sorgulanacak fatura numarası Invoice_id alanına yazılır.

The screenshot shows the Postman interface for the 'PlatformOde Api Entegrasyonu' workspace. A POST request is configured to the endpoint '({base_url})/api/checkstatus'. The request body is a JSON object with the following fields: 'invoice_id' (Abcd123456) and 'merchant_key' (a placeholder). The response body is a JSON object with the following fields: 'status_code' (69), 'status_description' (Order is not processed yet!), 'transaction_status' (Pre-Authorization), 'order_id' (VP17135180952054115), 'transaction_id' (Sqzuv-cg-TC10-37878-190424), 'message' (Order is not processed yet!), 'reason' (User did not complete the process), 'bank_status_code' (Approved(171351809517042)), 'bank_status_description' (Approved(171351809517042)), 'invoice_id' (Abcd123456), 'total_refunded_amount' (0), 'product_price' (26), 'transaction_amount' (26), 'ref_number' (empty), 'transaction_type' (Pre-Authorization), 'original_bank_error_code' (empty), 'original_bank_error_description' (empty), 'merchant_commission' (0.00), 'user_commission' (0.00), and 'settlement_date' (2024-04-19).

Ayrıca işlem durumu Üye İşyeri Panelinden aşağıdaki gibi sonuç izlenebilir.

Detaylar	İşlem ID	Sipariş Numarası	Durum	Kullanıcı Adı	Tutar	Ürün Tutarı	IP	Tarih & Saat	Link ID	GSM Numarası
Detaylar	Sqzuv-cg-TC10-37878-190424	VP17135180952054115	Ön Provizyon	-	26.00 ₺	26.00 ₺	193.108.213.144	19.04.2024 - 12:14	0	-
Detaylar	6LdKK-bg-TC10-37878-190424	VP17135090075482952	İade edildi	-	25.00 ₺	25.00 ₺	193.108.213.144	19.04.2024 - 09:43	0	-
İade Detayları										

Sayfa başına 10



9. Confirm Payment

Ödeme işleminin durumu değiştirmek için [Confirm Payment](#) linkine tıklanır. İşlem durumuna onay vermek için status=1, iptal için status=2 yapılabilir.

PlatformODE Api Entegrasyonu / Confirm Payment

POST {{base_url}}/api/confirmPayment

Params Authorization Headers (11) Body Pre-request Script Test

none form-data x-www-form-urlencoded raw binary Grap

```
1 {
2   "total": "34.00",
3   "invoice_id": "Abcd1234567",
4   "status": "1",
5   "merchant_key": "{{merchant_key}}",
6   "hash_key": "{{hash_key}}"
7 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status_code": 100,
3   "status_description": "The transaction has been Approved",
4   "transaction_status": "Completed",
5   "order_id": "VP17135261812340912",
6   "invoice_id": "Abcd1234567"
7 }
```

Detaylar	İşlem ID	Sipariş Numarası	Durum	Kullanıcı Adı	Tutar	Ürün Tutarı	IP	Tarih & Saat	Link ID	GSM Numarası
Detaylar	gbsmP-dg-TC10-37878-190424	VP17135261812340912	Tamamlanmış Ön Provizyon	-	34.00 ₺	34.00 ₺	193.108.213.144	19.04.2024 - 14:54	0	-
Detaylar	F0mwT-ag-TC10-37878-190424	VP17135262244581349	Tamamlandı	-	35.00 ₺	35.00 ₺	193.108.213.144	19.04.2024 - 14:30	0	-
Detaylar	SqzuV-cg-TC10-37878-190424	VP17135180952054115	Ön Provizyon	-	26.00 ₺	26.00 ₺	193.108.213.144	19.04.2024 - 12:14	0	-
Detaylar	6LdKX-bg-TC10-37878-190424	VP17135090075462952	İade edildi	-	25.00 ₺	25.00 ₺	193.108.213.144	19.04.2024 - 09:43	0	-
İade Detayları										



10. Get Installment

Tanımlanmış taksit sayısını görmek için [Get Installment](#) linkine tıklanır.

The screenshot displays the API Network interface. On the left, under 'My Workspace', the 'PlatformOde Api Entegrasyonu' collection is expanded, and the 'POST Get Installment' endpoint is highlighted with a red circle. The main panel shows the details for this endpoint, including the method 'POST', the URL '({{base_url}})/api/installments', and the body type 'raw'. The body content is a JSON object with a 'merchant_key' field. The 'Test Results' tab is active, showing a successful response with a status code of 100, a message 'Merchant Active Installment', and a list of installments (1, 2, 3, 4, 5, 6).

```
POST {{base_url}}/api/installments
```

Params Authorization Headers (13) Body Pre-request Script Tests Set

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1 {
2   "merchant_key": "{{merchant_key}}"
3 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status_code": 100,
3   "message": "Merchant Active Installment",
4   "installments": [
5     1,
6     2,
7     3,
8     4,
9     5,
10    6
11  ]
12 }
```



11. Get Pos

Kart numarasına göre taksit listesini görmek için [Get Pos](#) linkine tıklanır. Bin kodu ve taksit tutarı girilmesi gerekir.

The screenshot shows the Postman interface with the 'Get Pos' endpoint selected in the 'PlatformOde Api Entegrasyonu' collection. The endpoint is a POST request to `{{base_url}}/api/getpos`. The request body is a JSON object with the following fields:

```
1 {
2   "credit_card": "540061",
3   "amount": "100.00",
4   "currency_code": "TRY",
5   "merchant_key": "{{merchant_key}}"
6 }
```

The response is a JSON object with the following fields:

```
1 {
2   "status_code": 100,
3   "status_description": "Successfull",
4   "data": [
5     {
6       "pos_id": 1,
7       "campaign_id": 0,
8       "allocation_id": 0,
9       "installments_number": 1,
10      "card_type": "CREDIT CARD",
11      "card_program": "WORLD",
12      "card_scheme": "mastercard",
13      "payable_amount": 100,
14      "hash_key": "ff26f961c9e43802:b5be:EUoh__PaZWuaauv6z8CY__z1zoD8RIIc5BWwgA2CkpsPP68P0nm16AE07ooX84Mk96WysNX6JzjvP",
15      "amount_to_be_paid": "100.00",
16      "currency_code": "TRY",
17      "currency_id": 1,
18      "title": "Single payment"
19    }
20 ]
21 }
```




12. Merchant Commissions

Üye işyeri için tanımlanmış komisyon bilgisini görmek için [Merchant Commissions](#) linkine tıklanır.

The screenshot displays the API Network interface. In the sidebar, under 'My Workspace', the 'Collections' section is expanded, showing a list of endpoints under 'PlatformOde Api Entegrasyonu'. The 'POST Merchant Commissions' endpoint is highlighted. The main panel shows the details of the 'POST' request to '{{base_url}}/api/commissions'. The 'Body' tab is selected, showing a JSON response with status '100' and 'successful'. The response data includes details for 'Regular Commission', 'Save Card Token Payment Commission', and 'Foreign Card Commission'.

```
1 {
2   "status_code": 100,
3   "status_description": "successful",
4   "data": {
5     "1": [
6       {
7         "title": "Regular Commission",
8         "card_program": "CardFinans",
9         "merchant_commission_percentage": "0.0000",
10        "merchant_commission_fixed": "0.0000",
11        "user_commission_percentage": 0,
12        "user_commission_fixed": 0,
13        "currency_code": "TRY",
14        "installment": 1,
15        "pos_id": 1
16      },
17      {
18        "title": "Save Card Token Payment Commission",
19        "card_program": "",
20        "merchant_commission_percentage": "9",
21        "merchant_commission_fixed": "10",
22        "user_commission_percentage": "-",
23        "user_commission_fixed": "-",
24        "currency_code": "TRY",
25        "installment": 1
26      },
27      {
28        "title": "Foreign Card Commission",
29        "card_program": "",
30        "merchant_commission_percentage": "5",
31        "merchant_commission_fixed": "6",
```



13. Get Transactions

İstenilen tarihteki yapılan işlemleri listelemek için [Get Transactions](#) linkine tıklanır. Fatura numarası tercihe bağlı olup isteğe eklenirse girilen tarih ve fatura numarası ile başlayan işlemler listelenir.

The screenshot shows the Postman API client interface. On the left, the 'My Workspace' tab is active, and the 'PlatformOde Api Entegrasyonu' collection is expanded. The 'POST Get Transaction' endpoint is selected. The main panel displays the request details for the 'POST /api/getTransactions' endpoint. The request body is in JSON format, containing fields like 'merchant_key', 'hash_key', 'date', and 'invoiceid'. The response body is also shown in JSON format, displaying transaction details such as 'status_code', 'status_description', and a list of transactions with various fields like 'merchant_id', 'payment_id', 'gross', 'net', 'product_price', etc.

```
1 {
2   "merchant_key": "{{merchant_key}}",
3   "hash_key": "{{hash_key}}",
4   "date": "2024-04-17",
5   "invoiceid": "Abcd1245671" // Optional
6 }
```

```
1 {
2   "status_code": 100,
3   "status_description": "Successfull",
4   "data": {
5     "transactions": [
6       {
7         "merchant_id": 37878,
8         "payment_id": "bUeCY-ef-TC10-37878-170424",
9         "merchant_commission": "0",
10        "gross": "22",
11        "net": "22",
12        "payment_type_id": 1,
13        "product_price": "22",
14        "total_refunded_amount": "0",
15        "card_holder_bank": "ZİRAAT BANKASI",
16        "dpl_id": 0,
17        "currency_id": 1,
18        "invoice_id": "Abcd1245671",
19        "credit_card_no": "413226****0003",
20        "installment": 1,
21        "rolling_amount": "0",
22        "gsm_number": "",
23        "card_program": "BANKKART_COMBO",
24        "result": "Approved(1713341179572)",
25        "settlement_date_merchant": "2024-04-18 11:00:00",
26        "created_at": "2024-04-17T08:06:20.000000Z",
27        "updated_at": "2024-04-17T08:06:20.000000Z",
28        "recurring_id": 0,
29        "order_id": "41322611703706046"
30      }
31    ]
32  }
```



14. All Transactions

Bütün yapılan işlemleri listelemek için All Transactions linkine tıklanır.

PlatformOde Api Entegrasyonu / All Transaction

POST {{base_url}}/api/alltransaction

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "merchant_key": "{{merchant_key}}"
3 }
```

Body Cookies (1) Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1888 {
1889   "merchant_bank_iban": "TR130806291936737395624491",
1890   "merchant_bank_account": "23123",
1891   "remote_payment_method": "",
1892   "remote_internal_merchant_order_id": "",
1893   "dpl_receiver_mail": "",
1894   "bank_terminal_id": "",
1895   "dpl_dynamic_form_data": "",
1896   "product_name": "Item3"
1897 },
1898 "first_page_url": "https://testapp.platfomode.com.tr/ccpayment/api/alltransaction?page=1",
1899 "from": 1,
1900 "last_page": 5,
1901 "last_page_url": "https://testapp.platfomode.com.tr/ccpayment/api/alltransaction?page=5",
1902 "links": [
1903   {
1904     "url": "",
1905     "label": "&laquo; Previous",
1906     "active": false
1907   },
1908   {
1909     "url": "https://testapp.platfomode.com.tr/ccpayment/api/alltransaction?page=1",
1910     "label": "1",
1911     "active": true
1912   },
1913   {
1914     "url": "https://testapp.platfomode.com.tr/ccpayment/api/alltransaction?page=2",
1915     "label": "2",
1916     "active": false
1917   },
1918   {
1919     "url": "https://testapp.platfomode.com.tr/ccpayment/api/alltransaction?page=3",
```



15. Payment Complete

Ön Provizyon ile satış yapmak için Payment Complete linkine tıklanır.

Bu servise istek atarken payment_completed_by ibaresini "merchant" şeklinde yollarsanız, PaymentComplete servisine istek atıp ödemeyi tamamlamanız gerekiyor.